AFRL-IF-RS-TR-2006-27
Final Technical Report
January 2006

ALGORITHMS FOR FUNGIBLE COMPUTATION

Massachusetts Institute of Technology

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2006-27 has been reviewed and is approved for publication

APPROVED: /s/

THOMAS E. RENZ
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS
Deputy Chief, Advanced Computing Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** JANUARY 2006	**3. REPORT TYPE AND DATES COVERED** Final Jun 2003 – Sep 2005

4. TITLE AND SUBTITLE
ALGORITHMS FOR FUNGIBLE COMPUTATION

5. FUNDING NUMBERS
C - F30602-03-2-0090
PE - 31011G
PR - AFCM
TA - IT
WU - 03

6. AUTHOR(S)

Neil Gershenfeld

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/IFTC
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2006-27

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Thomas E. Renz/IFTC/(315) 330-3423 Thomas.Renz @rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This project sought fundamental mathematical principles to guide the development of “fungible” computers. Like fungible goods in economics that can be incrementally extended and exchanged, this effort sought to solve problems by using distributed processing nodes propagating mobile code through short-range communications. Motivations for doing this included providing extensibility for hardware to be able to scale with demand, offering reliable operation with unreliable components as process technologies reach fundamental limits, taking advantage of the structure of inherently distributed problems, and dynamically balancing the distribution of computation, storage, sensing, display, and I/O in an operating system. Results included proof-of-principle demonstration applications such as a display that operates statistically, and a unifying research framework based on “mathematical programming” that lies at the intersection of physically-inspired local algorithms, graphical message-passing, and global constrained optimization.

14. SUBJECT TERMS
Fungible Computing, Polymorphous Computing, Distributed Computing, Paintable Computing

15. NUMBER OF PAGES
18

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

1.0 Introduction

Today's prevalent computer programming model follows from work that was done on early scalar processors, and is no longer adequate for the challenges and opportunities presented by technological scaling, including the need for:

- incrementally adding resources to match the workload of an operating system
- building reliable systems from unreliable components in emerging process technologies
- taking advantage of distributed structure in an application
- integrating and optimizing a mix of sensing, actuation, display, and communications with computation
- packaging computation into embedded and conformal form factors

And ultimately the most severe limit of all is complexity itself, as the cost for taping out a chip and verifying a line of code in large-scale systems diverges faster than any other expense. This project sought to address these issues through the development of models for computation that can function like “fungible” goods in economics, which can be incrementally extended and exchanged.

One alternative programming approach appeals by analogy to the distributed dynamics of physical systems in creating mobile code for “paintable” computers [*Butera*], finding node-to-node distances by modeling chemical gradients, or tessellating an area with the laws of springs and masses. However, it remains difficult to understand the scaling behavior of algorithms that mimic physics, and to understand the quality of the local minima upon which the dynamics converge. Furthermore, there is no general principle from which we can map a problem statement on a distributed network to a physical system whose dynamics solve the problem.

Message-passing is another language for distributed problem-solving, factoring a problem into the propagation and transformation of local beliefs for the global solution [*Yedida et al.*]. This decomposition is exact on a tree, but leads to poorly characterized approximations for loopy graphs. The assumptions underlying message-passing algorithms have limited its use to specialized domains such as inference and high-rate coding.

The most successful approach for many of the largest and most intractable computational problems has been convex relaxations, lifting a mathematical program into a larger space that enforces global convexity, which can then efficiently be solved in polynomial time by interior-point or primal-dual methods [*Boyd and Vandenberghe*]. A growing variety of problems have been posed and solved in this way, offering tight bounds on the exact solution, and providing a principled way to understand what had been ad-hoc heuristics. While convex optimization does not directly address distributed implementations, sparse solvers do take advantage of structure in a problem.

The goal of this research was to develop the intersection of these programming approaches, capturing the intuitive dynamics of physical models while retaining the optimality of convex relaxations and the locality of message-passing. The framework that emerged, representing the Lagrange multipliers of a variational calculation in local dynamics, can be understood as using mathematical programs (optimization over goals with constraints) as a language for, rather than application of, computer programs [Fig. 1]. This approach emerged from a combination of theoretical studies, simulation, and hardware implementation.

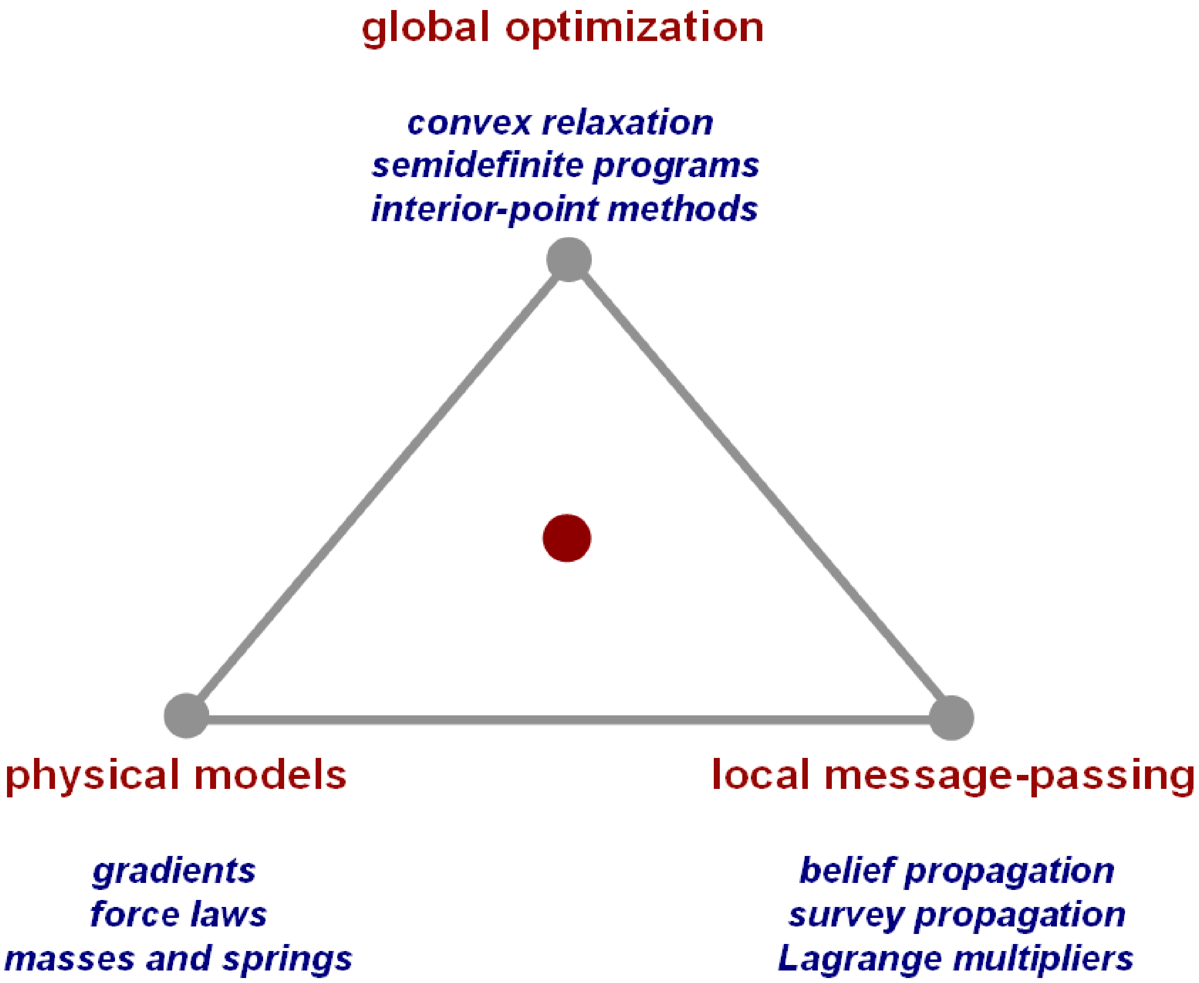


Figure 1: Mathematical programming

2.0 Results and Discussion

Summary

First Quarter:

- Hardware implementation of Postscript primitives for a paintable computer
- Convex relaxations of clustering
- Internet 0 event with the Internet's architects

Second Quarter:

- Experimental design for quantum optics, mixed-signal CMOS, and embedded networking applications
- Estimation of observation functions based on physical constraints
- Development of distributed dual decomposition algorithms

Third Quarter:

- Implementation of distributed dual decomposition algorithms
- Analog logic circuits for code acquisition and tracking

Fourth Quarter:

- Initial implementation of a paintable computing simulator to support algorithm and hardware development
- Development of SEA, the Scalable Encryption Algorithm for Small Embedded Applications.

First Quarter

The initial investigations addressed hardware implementation, software algorithms, and system networking.

Implementation

ARDA's algorithm support, along with complementary hardware support from DARPA, led to the development of a 1000 node prototype paintable computer. This was intended to serve as a guide for scaling to silicon, and as a debugging platform.

Each of the 1000 “pushpins” provided 60 MIPS and communicated locally through an optical link [Fig. 2]. Earlier simulated paintable computing algorithms were realized in hardware for the first time, passing mobile code through local shared memory. The first application was a minimal Postscript interpreter for a paintable computer. The great insight in Postscript was to recognize that a page should be described as an algorithm rather than a bitmap, converting it to pixels in a Raster Image Processor (RIP) in the output device. By analogy, we implemented an interpreter that converted Postscript to mobile code, which was turned into pixels in the display. The very same postscript that can be interpreted in a graphical previewer can be sent into the pushpins, where it will take advantage of the available nodes [Fig. 3]. Because of the effective integration of the I/0, frame buffer, and display elements, this display is fault-tolerant (nodes can be added and removed while it's running) and scalable (the image can grow as nodes are introduced).

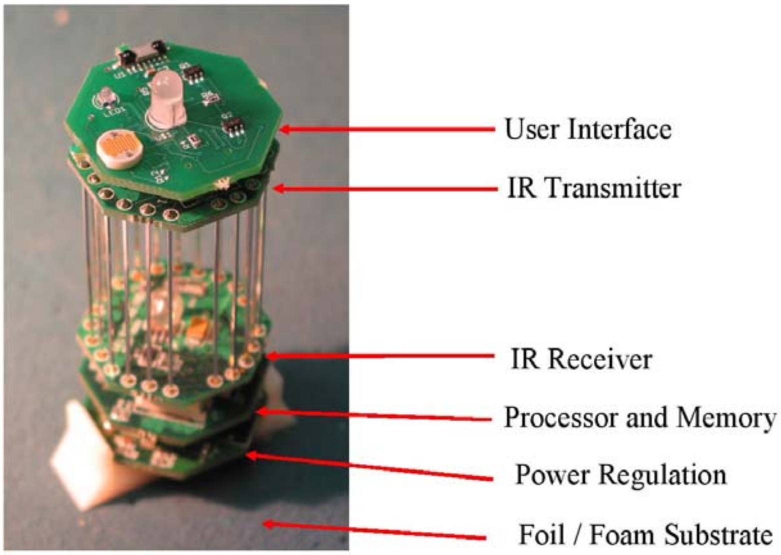
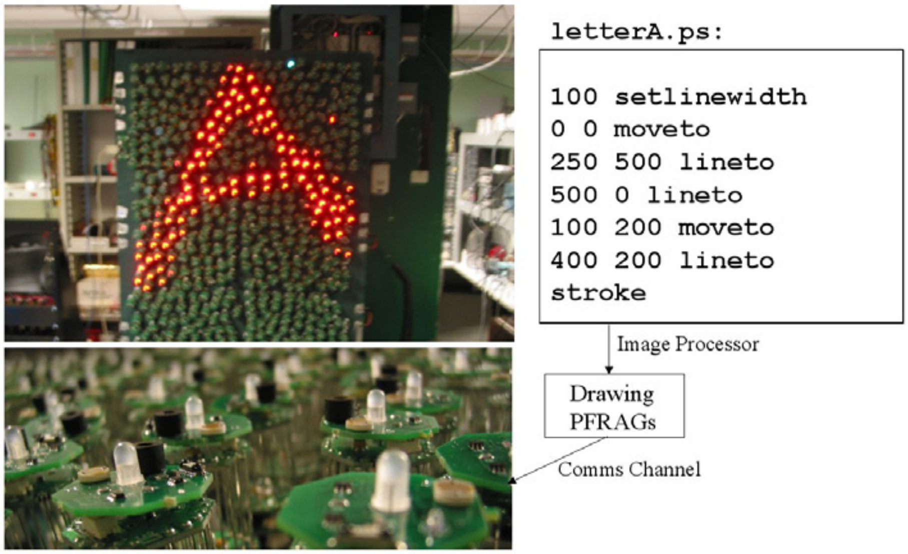


Figure 2: Paintable computing display node


```
letterA.ps:

100 setlinewidth
0 0 moveto
250 500 lineto
500 0 lineto
100 200 moveto
400 200 lineto
stroke
```

Figure 3: Paintable computing display

Algorithms

Distributed inference is an important prospective paintable computing application. In statistical learning, one is presented with data points paired to their associated function values and asked to infer the global hidden function. In regression, the function is real valued. In classification, the hidden function would be the mapping which labels data as belonging to some set. For example, this could be gene array sequences labeled “cancer” or “not cancer” or could be credit card transactions labeled “fraud” or “not fraud.”

If some, or most, of the function values are missing, such a learning problem is called “semi-supervised.” Not surprisingly, the semi-supervised learning problem is much harder than the fully supervised problem. Even when the problem statement is well posed, many regression problems become NP-Hard when the function values are hidden. We developed new algorithmic techniques for solving semi-supervised learning problems by appealing to the cost functions of the supervised learning problem.

Semi-supervised classification is often called clustering. We have developed a principled method of deriving clustering algorithms from classification algorithms and developed several relaxations for approximating the resulting clustering problems. We also reinterpreted currently popular clustering algorithms including the Normalized Cut algorithm of Shi and Malick as relaxations of the support vector machine classification engine with hidden labels. Using this interpretation of the Normalized Cut algorithm, we identified some of its weaknesses, and proposed new algorithms which don't suffer from these sensitivities.

Manifold learning, uncovering low dimensional structure in high dimensional data, can be interpreted as a semi-supervised regression problem where the map to be learned is the coordinate chart of the manifold. By having a plausible dynamic model which gives rise to the low dimensional motion embedded in a high dimensional space, we were able to learn these charts for a variety of embeddings. Extending techniques from SLAM (simultaneous location and mapping) to this highly nonlinear setting
and using state-of-the-art regression methods, we used dynamical models to supervise the learning and beat out most current methods of so-called “manifold learning.”

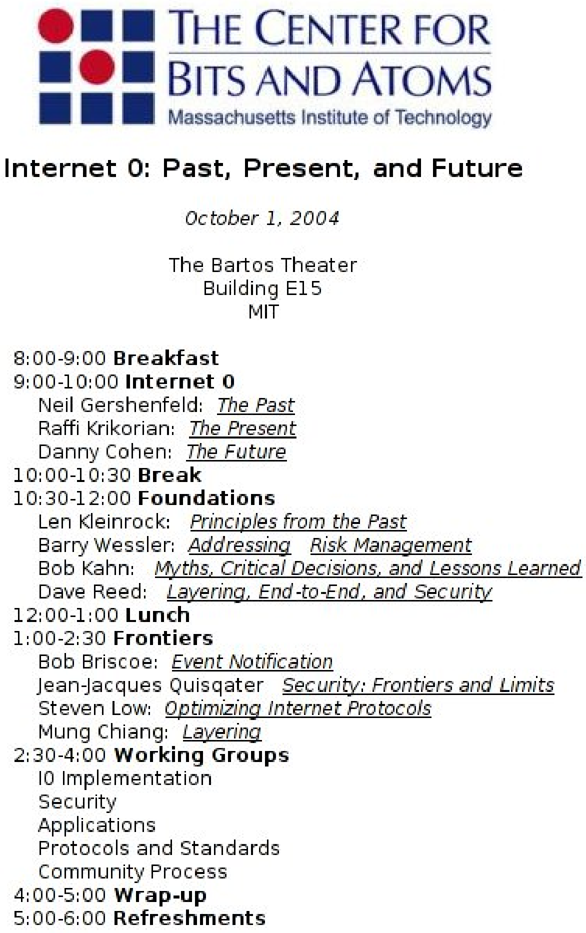
Networking

Along with paintable computing, the “Internet 0” architecture for interdevice internetworking emerged as an important application area for distributed algorithms in random systems, with many of the same goals and applications occurring over longer length scales [*Gershenfeld et al.*]. Support under this award contributed to running an event that gathered together the original Internet architects along with their current counterparts [Fig. 4].

This event led to two conclusions:

- A community process will be started to standardize and commercialize Internet 0, coordinated by industry partners

- An academic collaboration will be started on deriving networking protocols as distributed optimizations, using Internet 0 as a testbed



THE CENTER FOR BITS AND ATOMS
Massachusetts Institute of Technology

Internet 0: Past, Present, and Future

October 1, 2004

The Bartos Theater
Building E15
MIT

8:00-9:00 **Breakfast**
9:00-10:00 **Internet 0**
Neil Gershenfeld: *The Past*
Raffi Krikorian: *The Present*
Danny Cohen: *The Future*
10:00-10:30 **Break**
10:30-12:00 **Foundations**
Len Kleinrock: *Principles from the Past*
Barry Wessler: *Addressing* *Risk Management*
Bob Kahn: *Myths, Critical Decisions, and Lessons Learned*
Dave Reed: *Layering, End-to-End, and Security*
12:00-1:00 **Lunch**
1:00-2:30 **Frontiers**
Bob Briscoe: *Event Notification*
Jean-Jacques Quisqater *Security: Frontiers and Limits*
Steven Low: *Optimizing Internet Protocols*
Mung Chiang: *Layering*
2:30-4:00 **Working Groups**
I0 Implementation
Security
Applications
Protocols and Standards
Community Process
4:00-5:00 **Wrap-up**
5:00-6:00 **Refreshments**

Figure 4: Internet 0 agenda

Second Quarter

The work in this quarter progressed in three directions: hardware implementations, initial applications, and fundamental algorithms:

Implementations

Following the proof-of-principle demonstration of a PCB-scale paintable computer, the hardware effort progressed to a design study for scaling to silicon. This will require new process development in areas including die singulation, embedded and external power distribution, and die-to-die communication mechanisms. For a benchmark display application the cost scaling numbers appear to be competitive once the die shrink below about a millimeter, corresponding to about a 100nm feature size.

Beyond paintable computing, algorithm progress under this award guided an expansion of the associated experimental effort to study the compilation of mathematical programs into physical dynamics in a range of distributed systems. These included quantum optics (generating and detected coded light through the design of an ultra-fast laser cavity), mixed-signal CMOS (integrating the functions of a LNA, A/D and DSP to use the algorithmic structure in a weak signal to detect it in the presence of interference that would otherwise saturate the front end of a conventional receiver), and lightweight "Internet 0" network nodes (distributed naming and routing). The focus of the supporting algorithm work was the development of a "device driver" that could map the solution of a mathematical program onto the available degrees of freedom of a physical system.

Applications

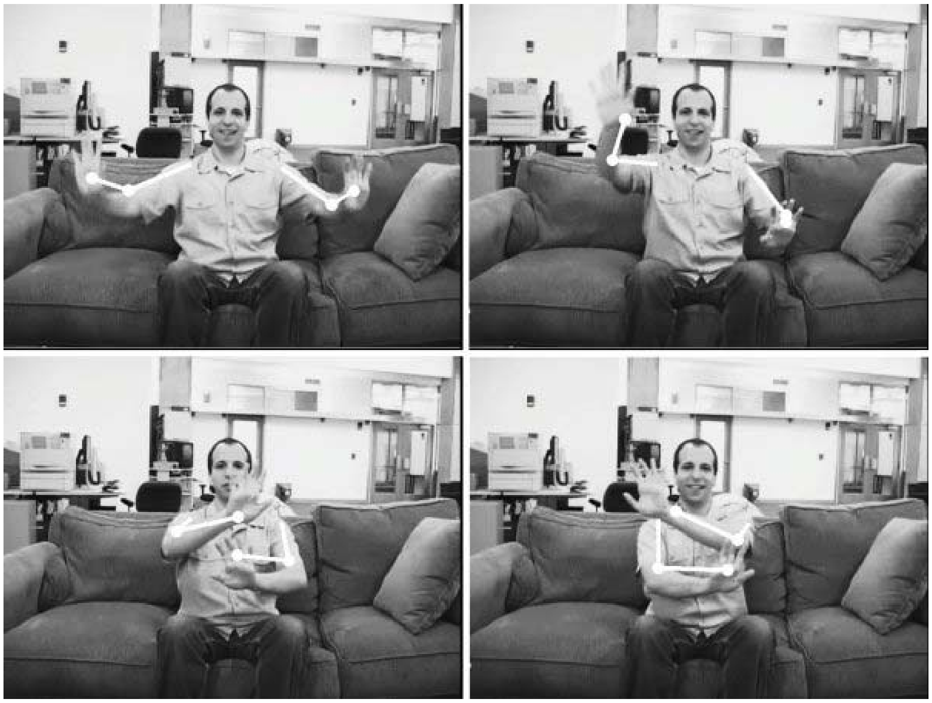


Figure 5: Conditioning on physics

Investigation of distributed solutions to constrained optimizations in the context of sensor fusion problems led to an unexpected application in image processing [*Rahimi et al.*]. We found that it was possible to estimate an observation function by conditioning a variational calculation on the constraint that the internal dynamics were governed by a low-dimensional physical process. This prior allows for semi-supervised learning in which the coordinates for a small set of extremal configurations are fixed and then the mapping is inferred from an otherwise arbitrary unlabeled data set. This allows for generalization far beyond the limits of an approach based on conventional functional approximation, and has a natural distributed least-squares implementation for motion tracking applications. The algorithm is shown here inferring the three-dimensional motion of arms from two-dimensional images [Fig. 5].

Algorithms

Practical techniques to solve sparse semidefinite programs in a distributed manner have been elusive as all efficient solution methods require global second derivative information. On the other hand, first-order methods require only local information but lack guaranteed complexity bounds. In revisiting classical first order methods such as the Dantzig-Wolfe dual decomposition, we investigated how to construct fast distributed methods with guaranteed performance bounds. In the limit of a large number of interacting systems, these methods are guaranteed to converge in a stable manner to an approximate optimal solution.

Third Quarter

We developed a methodology for generating fast algorithms for designing, modeling, and analyzing complex systems, which have no local minima, respect problem structure, and are provably reliable. This merged techniques from convex analysis related to classical methods such as the Dantzig-Wolfe decomposition. After formulating a problem as goals and constraints, our approach solved the dual problem using a distributed decomposition method. The dual problem is always convex and hence has no local minima, and there are a variety of available decompositions whose update rules respect the distributed nature of the constraints and costs in the problem.

We found broad applicability of these techniques. For paintable computing, we developed a general distributed method for consensus problems such as distributed tracking, object recognition, and decision theory. We applied this methodology to develop novel methods for decoding algorithms and carrier acquisition in RF circuits. We then investigated how to construct networking protocols by describing network goals as optimization problems and using decomposition methods to solve them, with a focus on Internet 0 as an application domain.

The extension of mathematical programming ideas to RF electronics progressed with the development of both baseband and microwave circuits for spread-spectrum acquisition and tracking [Fig. 6].

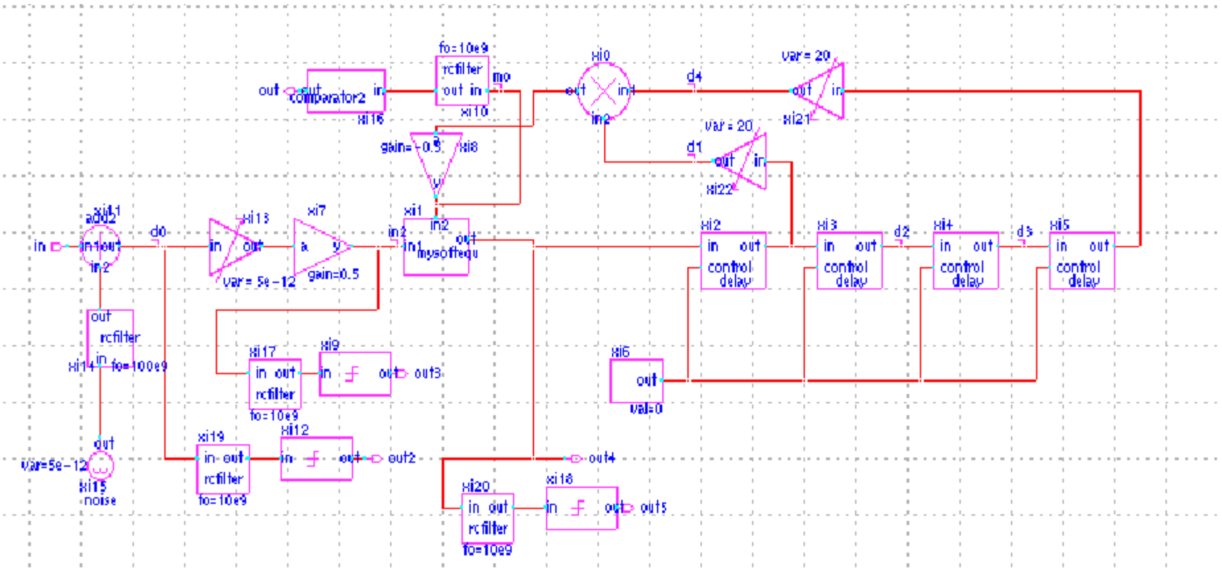


Figure 6: Noise-Locked Loop

The measured performance is shown here versus a comparator in recovering a noisy spread-spectrum signal [Figure 7].

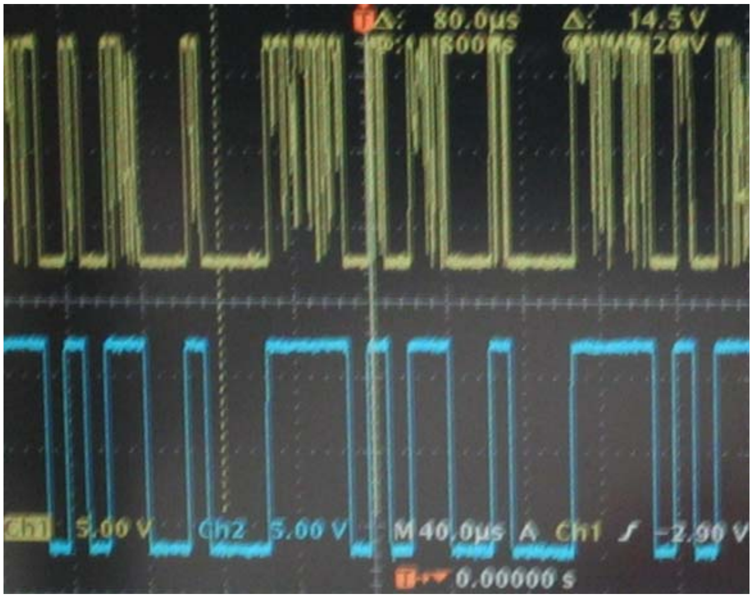


Figure 7: NLL spread-spectrum recovery

Potential benefits of this approach include reduced power consumption and parts count, higher clock speeds, and improved interference rejection by bit-slicing at the algorithm's output rather than its input.

Fourth Quarter

The transition from the proof-of-principle demonstrations under this award to next-generation hardware and eventual silicon scaling required a software simulator that reflects progress in programming models. An initial implementation was done of a second-generation simulator based on passing and transforming mathematical strings rather than the earlier assumption of fragments of conventional code. In this quarter, a prototype of string passing, scheduling and evaluation was written. Four subsequent stages are anticipated in the future development of this simulator:

- First, it will be used for algorithm design and testing
- Then, this will be ported to an MPI platform for scaling studies of algorithm performance beyond the limits of modeling on scalar machines
- Next, the simulator will be instrumented to extract physical parameters of interest such as power consumption and communication performance, to guide hardware design with conventional processors
- Finally, the functional representation used in the simulator will be migrated to develop devices that use this natively

Given the sensitivity and importance of many of the applications of paintable computing, it will be important to provide security at the level of communications between individual nodes. However, existing cryptosystems for embedded devices either impose significant hardware requirements or offer limited cryptographic protection. To meet this need, the Scalable Encryption Algorithm for Small Embedded Applications (SEA) was developed [*Standaert et al.*]. It is based on a Feistel cipher, reduced to a minimal set of primitive operations available in resource-constrained processors: bitwise XOR, bit and word rotation, modular addition, and a 3-bit S-box as the nonlinear element. Advantages of SEA include:

- Scalability across word and memory sizes
- Provable security against linear and differential cryptanalysis
- Provable diffusion
- Symmetrical encryption and decryption
- On-the-fly key schedule derivation
- Tradeoff from space to time

A first implementation over an Internet 0 transport was realized in just 386 bytes of code [Fig. 8].

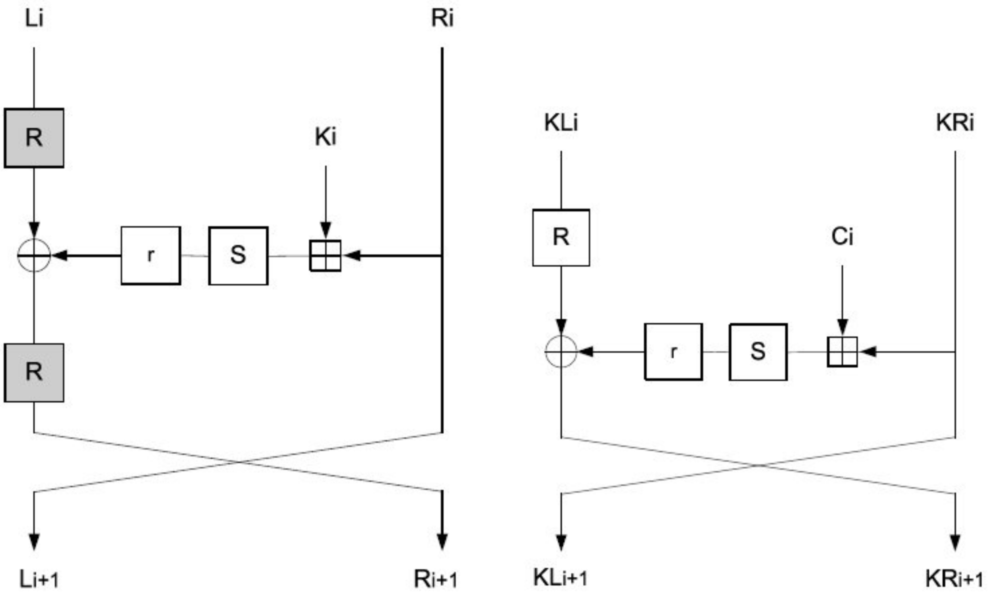


Figure 8: Scalable Encryption Algorithm encrypt/decrypt and key rounds

3.0 Conclusions

This project accomplished its original aim of demonstrating, through both software simulation and hardware implementation, algorithms that can incrementally scale with computational resources, most notably building a display that operates statistically and is capable of distributed rendering. The project also resulted in industrial technology transfer through both scaling of paintable computing and commercialization of Internet 0. And it provided a theoretical foundation for future work, based on compiling a problem posed as a mathematical program with global goals and constraints into a variational calculation that introduces Lagrange multipliers that get passed locally. In the appropriate limits this approach can recover both message-passing algorithms and physically-inspired models, while retaining the performance and bounds of global optimization.

Even more important, and unexpected, was the recognition that these same ideas are equally applicable for assembling devices into circuits, circuits into systems, and systems into networks, with the Lagrange multipliers being carried by voltages, strings, and packets respectively. This can be understood as the reverse of the formal structure of statistical mechanics, in order to prescribe rather than describe the behavior of enormously complex systems [Fig. 9]. Instead of starting from local dynamics, imposing a global extremal principle (maximum entropy), and then predicting physical observables from Lagrange multiplies introduced by constraints (such as temperature for fixed average energy), a global goal is imposed by a problem, the constraints associated with its implementation introduce Lagrange multipliers that are then carried locally to solve the global problem.

That possibility suggests that this research may help lead to not just paintable computers but fungible information technologies at all levels of description. Beyond the technical challenges and opportunities that this observation presents, these questions lie at the interface between hardware and software, and between applications and their implementation. They do not fit within the conventional boundaries between activities in these areas, providing an opportunity for future work to cross them.

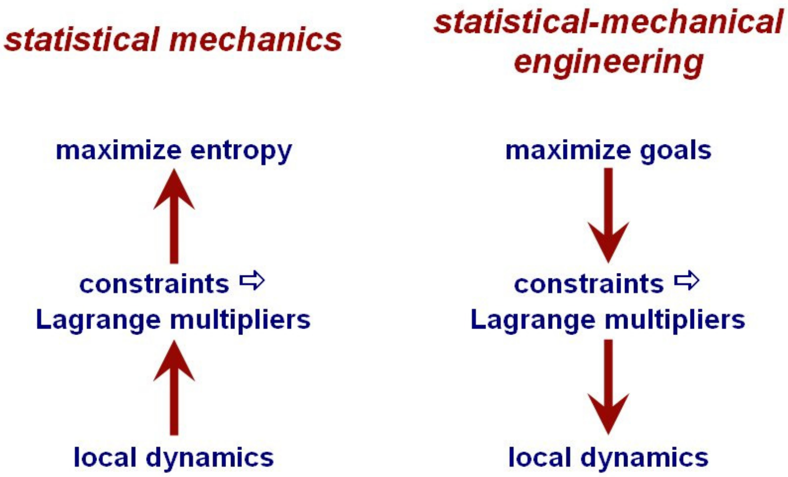

statistical mechanics
statistical-mechanical engineering
maximize entropy
maximize goals
constraints ⇨ Lagrange multipliers
constraints ⇨ Lagrange multipliers
local dynamics
local dynamics

Figure 9: Statistical-mechanical engineering

4.0 Future Work

This research has provided both theoretical direction for, and experimental demonstration of, paintable computing. However, designing and building full-custom silicon for a complete system based on this work, including subsystems for computing, communications, power, and transduction, would represent a significant financial and time commitment.

Instead, it's likely to be faster, cheaper, and more useful to progress through a series of smaller projects that can advance, demonstrate, and evaluate key capabilities, including incremental extensibility, adaptive code execution in random configurations, and virtual and physical reconfigurability. These test cases could be chosen to be immediately useful as well as providing a technology base for a fundamentally fungible computing technology.

Such next-generation prototyping could be done with available state-of-the-art packaging, including sub-mm discretes and few-mm bare die. A particular focus should be on understanding the minimum local node complexity needed for global scalability, a key question that is likely to be relevant to a range of technologies beyond conventional silicon scaling, particularly early applications of molecular electronics.

The cost for such a follow-up project would be in the range of $500k-$1M per year, and involve a larger academic and industrial team of participants.

5.0 References

[*Butera*] W. Butera, *Programming a Paintable Computer*, Ph.D. thesis, MIT (2002)

[*Yedidia et al.*] J.S. Yedidia, W.T. Freeman, and Y. Weiss, *Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms*, IEEE Transactions on Information Theory, Vol. 51, Issue 7, pp. 2282-2312, (2005)

[*Boyd and Vandenberghe*] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, (2004)

[*Standaert et al.*] F.X.Standaert, G. Piret, N. Gershenfeld, and J.J. Quisquater, *SEA: a Scalable Encryption Algorithm for Small Embedded Applications*, ECRYPT Workshop on RFID and Lightweight Crypto, Graz, Austria (2005)

[*Gershenfeld et al.*] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen, *The Internet of Things*, Scientific American 291, pp. 76-81 (2004)

[*Rahimi et al.*] A. Rahimi, B. Recht and T. Darrell, *Learning Appearance Manifolds from Video*, Computer Vision and Pattern Recognition San Diego, CA (2005)